
xirr Documentation

Release 0.1.7.post1.dev3

Patrick Ruckstuhl

Mar 09, 2022

CONTENTS

| | | |
|----------|----------------------------|----------|
| 1 | Contents | 3 |
| 1.1 | Installation | 3 |
| 1.2 | License | 3 |
| 1.3 | xirr | 3 |
| 2 | Indices and tables | 5 |
| | Python Module Index | 7 |
| | Index | 9 |

Irregular internal rate of return (xirr) and net present value (npv) calculations.

Based on <https://stackoverflow.com/questions/8919718/financial-python-library-that-has-xirr-and-xnpv-function> with some ideas for handling special cases from <https://github.com/RayDeCampo/java-xirr/>

CONTENTS

1.1 Installation

As this is released on [PyPI](#) you can simply install it with

```
pip install xirr
```

1.2 License

The MIT License (MIT)

Copyright (c) 2021 Patrick Ruckstuhl

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.3 xirr

1.3.1 xirr package

Submodules

xirr.math module

`xirr.math.cleanXirr(valuesPerDate: dict[datetime.date, float])` → `Optional[float]`

A “cleaned” version of the xirr which avoids returning a xirr for some extreme cases and ignores amounts which are almost 0.

`xirr.math.listsXirr`(*dates*: *list*[*datetime.date*], *values*: *list*[*float*], *whichXirr*:
typing.Callable[[*dict*[*datetime.date*, *float*]], *typing.Optional*[*float*]] = <function xirr>) →
Optional[*float*]

A convenience function that takes two lists of dates and values rather than a combined dictionary.

Use `whichXirr` to select the actual xirr function to use.

Anti-pattern: Using a simple dictionary comprehension would not work, e.g. `xirr({d: v for d, v in zip(dates, values)})` Because this overwrites entries with identical dates.

`xirr.math.xirr`(*valuesPerDate*: *dict*[*datetime.date*, *float*]) → *Optional*[*float*]

Calculate the irregular internal rate of return.

```
>>> from datetime import date
>>> valuesPerDate = {date(2019, 12, 31): -80005.8, date(2020, 3, 12): 65209.6}
>>> xirr(valuesPerDate)
-0.645363882724717
```

`xirr.math.xnpv`(*valuesPerDate*: *dict*[*datetime.date*, *float*], *rate*: *float*) → *float*

Calculate the irregular net present value.

```
>>> from datetime import date
>>> valuesPerDate = {date(2019, 12, 31): -100, date(2020, 12, 31): 110}
>>> xnpv(valuesPerDate, -0.10)
22.257507852701295
```

Module contents

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

PYTHON MODULE INDEX

X

xirr, 4

xirr.math, 3

INDEX

C

`cleanXirr()` (*in module xirr.math*), 3

L

`listsXirr()` (*in module xirr.math*), 3

M

module

 xirr, 4

 xirr.math, 3

X

xirr

 module, 4

`xirr()` (*in module xirr.math*), 4

xirr.math

 module, 3

`xnpv()` (*in module xirr.math*), 4